State-of-the-art predictive and prescriptive analytics for IEEE CIS "predict+optimise" Challenge

Rasul Esmaeilbeigi

Joint work with: Mahdi Abolghasemi

December 05, 2021

Summary of the challenge and our results

- **Prediction:** forecasting fifteen minutely slots of energy load for six solar panels and six buildings
- **Optimisation:** scheduling of batteries and activities such that the total energy cost is minimised

Summary of the challenge and our results

- **Prediction:** forecasting fifteen minutely slots of energy load for six solar panels and six buildings
- **Optimisation:** scheduling of batteries and activities such that the total energy cost is minimised

- **Phase 1:**
  - Prediction: $MASE = 0.98$

Summary of the challenge and our results

- **Prediction:** forecasting fifteen minutely slots of energy load for six solar panels and six buildings
- **Optimisation:** scheduling of batteries and activities such that the total energy cost is minimised

- **Phase 1:**
  - Prediction: $MASE = 0.98$
  - Optimisation: $COST = 439,071$ (**1st** in the Leaderboard) ($\boldsymbol{avg.gap = 7.9\%}$)

Summary of the challenge and our results

- **Prediction:** forecasting fifteen minutely slots of energy load for six solar panels and six buildings
- **Optimisation:** scheduling of batteries and activities such that the total energy cost is minimised

- **Phase 1:**
  - Prediction: $MASE = 0.98$
  - Optimisation: $COST = 439,071$ (**1st** in the Leaderboard) ($\boldsymbol{avg.gap = 7.9\%}$)

- **Phase 2:**
  - Prediction: $MASE = 0.74$ (**2nd** in the Leaderboard)

Summary of the challenge and our results

- **Prediction:** forecasting fifteen minutely slots of energy load for six solar panels and six buildings
- **Optimisation:** scheduling of batteries and activities such that the total energy cost is minimised

- **Phase 1:**
  - Prediction: $MASE = 0.98$
  - Optimisation: $COST = 439,071$ (**1st** in the Leaderboard) ($\boldsymbol{avg.gap = 7.9\%}$)

- **Phase 2:**
  - Prediction: $MASE = 0.74$ (**2nd** in the Leaderboard)
  - Optimisation: $COST = 328,359$ (**1st** in the Leaderboard) ($\boldsymbol{avg.gap = 2.5\%}$)

Forecasting

- Started by exploring the data and looking for trends, pattern and seasonality

Forecasting

- Started by exploring the data and looking for trends, pattern and seasonality
- Selected LightGBM due to being fast and providing reliable forecasts

Forecasting

- Started by exploring the data and looking for trends, pattern and seasonality
- Selected LightGBM due to being fast and providing reliable forecasts
- Input variables
  - calendar features
  - daily weather data
  - hourly weather data
  - various rolling statistics of these features

Forecasting

- Started by exploring the data and looking for trends, pattern and seasonality
- Selected LightGBM due to being fast and providing reliable forecasts
- Input variables
  - calendar features
  - daily weather data
  - hourly weather data
  - various rolling statistics of these features

- Generated several forecasts with different models to provide multiple scenarios

Forecasting

- Started by exploring the data and looking for trends, pattern and seasonality
- Selected LightGBM due to being fast and providing reliable forecasts
- Input variables
  - calendar features
  - daily weather data
  - hourly weather data
  - various rolling statistics of these features

- Generated several forecasts with different models to provide multiple scenarios
- The final submitted forecasts is an ensemble of two LightGBM models where daily and hourly features were used and hyperparameters were optimised using grid search.

Optimisation

- Input data for each instance:
  - the parameters available in the ppoi file and the price values per time slot

Optimisation

- Input data for each instance:
  - the parameters available in the ppoi file and the price values per time slot
  - **net base load** (l_t): the total predicted base load of buildings minus generation of their solar panels (at time slot t)

Optimisation

- Input data for each instance:
  - the parameters available in the ppoi file and the price values per time slot
  - **net base load** (l_t): the total predicted base load of buildings minus generation of their solar panels (at time slot t)
- Optimisation under uncertainty (l_t is the uncertain)

Optimisation

- Input data for each instance:
  - the parameters available in the ppoi file and the price values per time slot
  - **net base load** (l_t): the total predicted base load of buildings minus generation of their solar panels (at time slot t)

- Optimisation under uncertainty (l_t is the uncertain)
- Mixed integer linear programming (MILP) techniques
  - mathematical model employs both continuous and binary decision variables
  - the problem conditions and requirements are represented as "linear" constraints

Optimisation

- Input data for each instance:
  - the parameters available in the ppoi file and the price values per time slot
  - **net base load** (l_t): the total predicted base load of buildings minus generation of their solar panels (at time slot t)

- Optimisation under uncertainty (l_t is the uncertain)
- Mixed integer linear programming (MILP) techniques
  - mathematical model employs both continuous and binary decision variables
  - the problem conditions and requirements are represented as "linear" constraints
  - the objective function must also be "linear" (e.g., no product or power of variables)

Optimisation

- Input data for each instance:
  - the parameters available in the ppoi file and the price values per time slot
  - **net base load** (l_t): the total predicted base load of buildings minus generation of their solar panels (at time slot t)

- Optimisation under uncertainty (l_t is the uncertain)
- Mixed integer linear programming (MILP) techniques
  - mathematical model employs both continuous and binary decision variables
  - the problem conditions and requirements are represented as "linear" constraints
  - the objective function must also be "linear" (e.g., no product or power of variables)

- Why MILP?
  - flexibility in capturing various requirements and adapting to the changes

Optimisation

- Input data for each instance:
  - the parameters available in the ppoi file and the price values per time slot
  - **net base load** (l_t): the total predicted base load of buildings minus generation of their solar panels (at time slot t)

- Optimisation under uncertainty (l_t is the uncertain)
- Mixed integer linear programming (MILP) techniques
  - mathematical model employs both continuous and binary decision variables
  - the problem conditions and requirements are represented as "linear" constraints
  - the objective function must also be "linear" (e.g., no product or power of variables)

- Why MILP?
  - flexibility in capturing various requirements and adapting to the changes
  - availability of highly advanced solvers
  - continuous improvement of such solvers

Optimisation

- Input data for each instance:
  - the parameters available in the ppoi file and the price values per time slot
  - **net base load** (l_t): the total predicted base load of buildings minus generation of their solar panels (at time slot t)

- Optimisation under uncertainty (l_t is the uncertain)
- Mixed integer linear programming (MILP) techniques
  - mathematical model employs both continuous and binary decision variables
  - the problem conditions and requirements are represented as "linear" constraints
  - the objective function must also be "linear" (e.g., no product or power of variables)

- Why MILP?
  - flexibility in capturing various requirements and adapting to the changes
  - availability of highly advanced solvers
  - continuous improvement of such solvers
  - solving to optimality or at least providing an optimality gap

Decision variables

Variable	Definition
x_{bt}	binary variable equal to 1 iff battery $b \in \mathcal{B}$ is charging at time $t \in \mathcal{T}$
y_{bt}	binary variable equal to 1 iff battery $b \in \mathcal{B}$ is discharging at time $t \in \mathcal{T}$
z_{at}	binary variable equal to 1 iff activity $a \in \mathcal{A}$ begins at time $t \in \mathcal{T}'_a$
v_{at}	binary variable equal to 1 iff activity $a \in \mathcal{A}$ is in progress at time $t \in \mathcal{T}_a$
s_{bt}	non-negative variable representing the state of battery $b \in \mathcal{B}$ at the end of time slot $t \in \mathcal{T}$
w_a	binary variable equal to 1 iff activity $a \in \mathcal{A}$ is scheduled
u_a	binary variable equal to 1 iff activity $a \in \mathcal{A}^O$ is scheduled outside working hours
d_a	non-negative variable representing the day index at which activity $a \in \mathcal{A}$ begins if it is scheduled
ℓ_t	unrestricted variable representing the aggregate/total load at time $t \in \mathcal{T}$
η	non-negative variable representing the absolute value of the maximum load
λ_i	binary variable equal to 1 iff $\lceil \eta \rceil$ is equal to $i \in \{1, \ldots, M\}$

Mathematical formulation

$$\min \quad \frac{0.25}{1000}\sum_{t\in\mathcal{T}}\pi_t\ell_t + 0.005\sum_{i=1}^{M} i^2\lambda_i - \sum_{a\in\mathcal{A}^O}(r_a w_a - p_a u_a) \tag{1}$$

$$\sum_{t'\in\mathcal{T}'_a\cap\{t-\delta_a+1,\ldots,t\}} z_{at'} = v_{at} \qquad a\in\mathcal{A}, t\in\mathcal{T}_a \tag{2}$$

$$\sum_{t\in\mathcal{T}_a} v_{at} = \delta_a w_a \qquad a\in\mathcal{A} \tag{3}$$

$$\sum_{t\in\mathcal{T}'_a} z_{at} = w_a \qquad a\in\mathcal{A} \tag{4}$$

$$\sum_{t\in\mathcal{T}''_a} z_{at} = u_a \qquad a\in\mathcal{A}^O \tag{5}$$

$$\sum_{t\in\mathcal{T}'_a}\left\lfloor\frac{t}{D}\right\rfloor z_{at} + \left\lceil\frac{T+1}{D}\right\rceil(1-w_a) = d_a \qquad a\in\mathcal{A} \tag{6}$$

$$d_a + w_a \le d_{a'} \qquad a'\in\mathcal{A}, a\in\mathcal{P}_{a'} \tag{7}$$

$$w_{a'} \le w_a \qquad a'\in\mathcal{A}, a\in\mathcal{P}_{a'} \tag{8}$$

$$s_{bt} = c'_b + 0.25\, m_b\,(x_{bt} - y_{bt}) \qquad b\in\mathcal{B}, t=1 \tag{9}$$

$$s_{bt} = s_{b,t-1} + 0.25\, m_b\,(x_{bt} - y_{bt}) \qquad b\in\mathcal{B}, t\in\mathcal{T}\setminus\{1\} \tag{10}$$

$$x_{bt} + y_{bt} \le 1 \qquad b\in\mathcal{B}, t\in\mathcal{T} \tag{11}$$

$$\sum_{a\in\mathcal{A}^O} n_a^L v_{at} + \sum_{a\in\mathcal{A}^R} n_a^L v_{a[t]} \le L \qquad t\in\mathcal{T} \tag{12}$$

$$\sum_{a\in\mathcal{A}^O} n_a^S v_{at} + \sum_{a\in\mathcal{A}^R} n_a^S v_{a[t]} \le S \qquad t\in\mathcal{T} \tag{13}$$

$$\begin{aligned}\ell_t = l_t &+ \sum_{b\in\mathcal{B}}\frac{m_b}{\sqrt{c_b}}(x_{bt} - e_b y_{bt}) \\ &+ \sum_{a\in\mathcal{A}^O}\beta_a(n_a^S + n_a^L)v_{at} \\ &+ \sum_{a\in\mathcal{A}^R}\beta_a(n_a^S + n_a^L)v_{a[t]} \qquad t\in\mathcal{T}\end{aligned} \tag{14}$$

$$\sum_{i=1}^{M}\lambda_i \le 1 \tag{15}$$

$$\sum_{i=1}^{M} i\lambda_i \ge \eta \tag{16}$$

$$\eta \ge \ell_t \qquad t\in\mathcal{T} \tag{17}$$

$$\eta \ge -\ell_t \qquad t\in\mathcal{T} \tag{18}$$

$$w_a = 1 \qquad a\in\mathcal{A}^R \tag{19}$$

$$0 \le s_{bt} \le c_b \qquad b\in\mathcal{B}, t\in\mathcal{T} \tag{20}$$

Objective function

$$\min \quad \frac{0.25}{1000} \sum_{t \in \mathcal{T}} \pi_t \ell_t + 0.005 \, \boldsymbol{\eta^2} \qquad - \sum_{a \in \mathcal{A}^O} (r_a w_a - p_a u_a)$$

Objective function

$$\min \quad \frac{0.25}{1000} \sum_{t \in \mathcal{T}} \pi_t \ell_t + 0.005\, \boldsymbol{\eta^2} \qquad - \sum_{a \in \mathcal{A}^O} (r_a w_a - p_a u_a)$$

$$\Downarrow$$

$$\min \quad \frac{0.25}{1000} \sum_{t \in \mathcal{T}} \pi_t \ell_t + 0.005 \sum_{\boldsymbol{i=1}}^{\boldsymbol{M}} \boldsymbol{i^2 \lambda_i} - \sum_{a \in \mathcal{A}^O} (r_a w_a - p_a u_a)$$

$$\sum_{i=1}^{M} \lambda_i \leq 1$$

$$\sum_{i=1}^{M} i\lambda_i \geq \eta$$

$$\eta \geq \ell_t \qquad t \in \mathcal{T}$$

$$\eta \geq -\ell_t \qquad t \in \mathcal{T}$$

Dealing with the uncertainty

- The deterministic formulation assumes that the true realisation of the net base load is given (l_t for $t \in \mathcal{T}$)

Dealing with the uncertainty

- The deterministic formulation assumes that the true realisation of the net base load is given (l_t for $t \in \mathcal{T}$)
- Although the forecasting methodology attempts to predict a good value for this parameter, the resulting forecast is always subject to an error
  - inherent randomness
  - the choice of the forecasting methodology

Dealing with the uncertainty

- The deterministic formulation assumes that the true realisation of the net base load is given (l_t for $t \in \mathcal{T}$)
- Although the forecasting methodology attempts to predict a good value for this parameter, the resulting forecast is always subject to an error
  - inherent randomness
  - the choice of the forecasting methodology
- We consider the net base load l_t as a random variable and use the so-called sample average approximation method
- We minimise the average cost of a solution over multiple scenarios (or forecasts) rather than just one
- This approach generally prescribes a solution with least expected cost that is also less sensitive to the forecasting errors.

The sample average approximation method

$$\min \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\left(\frac{0.25}{1000}\sum_{t\in\mathcal{T}}\pi_t\ell_{ts}+0.005\sum_{i=1}^{M}i^2\lambda_{is}\right)-\sum_{a\in\mathcal{A}^O}(r_a w_a-p_a u_a)$$

$$\begin{aligned}
\ell_{ts} &= l_{ts}+\sum_{b\in\mathcal{B}}\frac{m_b}{\sqrt{e_b}}(x_{bt}-e_b y_{bt}) \\
&\quad+\sum_{a\in\mathcal{A}^O}\beta_a(n_a^S+n_a^L)v_{at} \\
&\quad+\sum_{a\in\mathcal{A}^R}\beta_a(n_a^S+n_a^L)v_{a[t]} && s\in\mathcal{S}, t\in\mathcal{T} \\
\sum_{i=1}^{M}\lambda_{is} &\le 1 && s\in\mathcal{S} \\
\sum_{i=1}^{M}i\lambda_{is} &\ge \eta_s && s\in\mathcal{S} \\
\eta_s &\ge \ell_{ts} && s\in\mathcal{S}, t\in\mathcal{T} \\
\eta_s &\ge -\ell_{ts} && s\in\mathcal{S}, t\in\mathcal{T}
\end{aligned}$$

Algorithm design

- Started with solving the formulation without any enhancement techniques
- Gradually improved the algorithms
  - preprocessing (not scheduling some once-off activities outside working hours)
  - postprocessing (allocation of building rooms to activities to avoid symmetry and also reduce the problem size)
  - designing various fix-and-optimise algorithms to obtain a good solution quickly
  - warm-starting the solver

Algorithm design

- Started with solving the formulation without any enhancement techniques
- Gradually improved the algorithms
  - preprocessing (not scheduling some once-off activities outside working hours)
  - postprocessing (allocation of building rooms to activities to avoid symmetry and also reduce the problem size)
  - designing various fix-and-optimise algorithms to obtain a good solution quickly
  - warm-starting the solver
- Fix-and-optimise algorithms
  - relaxing the integrality constraints of the variables corresponding to the batteries
  - excluding the batteries (as if there are no batteries)
  - excluding penalized activities (once-off activities that result in a penalty)
  - restricting start time of all activities to even time slots

Implementation

- **Python 3.9 & Gurobi 9.1.2**

Implementation

- **Python 3.9 & Gurobi 9.1.2**
- **Default algorithm – with warm-start**
  1. relax the integrality constraints of battery (charge/discharge)
  2. solve the relaxed problem (0.9 of the time limit)
  3. fix the schedule for activities and reintroduce the removed integrality constraints
  4. solve the problem (0.1 of the time limit)

Implementation

- **Python 3.9 & Gurobi 9.1.2**
- **Default algorithm – with warm-start**
  1. relax the integrality constraints of battery (charge/discharge)
  2. solve the relaxed problem (0.9 of the time limit)
  3. fix the schedule for activities and reintroduce the removed integrality constraints
  4. solve the problem (0.1 of the time limit)
- **Default algorithm – no warm-start**
  1. exclude the batteries
  2. exclude the penalized activities
  3. restricting start time of all activities to even time slots
  4. solve the relaxed problem (0.7 of the time limit)
  5. fix the activity schedules (with some degrees of flexibility) in the original model
  6. solve the problem (0.3 of the time limit)

Implementation

- **Python 3.9 & Gurobi 9.1.2**
- **Default algorithm – with warm-start**
  1. relax the integrality constraints of battery (charge/discharge)
  2. solve the relaxed problem (0.9 of the time limit)
  3. fix the schedule for activities and reintroduce the removed integrality constraints
  4. solve the problem (0.1 of the time limit)
- **Default algorithm – no warm-start**
  1. exclude the batteries
  2. exclude the penalized activities
  3. restricting start time of all activities to even time slots
  4. solve the relaxed problem (0.7 of the time limit)
  5. fix the activity schedules (with some degrees of flexibility) in the original model
  6. solve the problem (0.3 of the time limit)
- **Our last submission used solutions of the previous submission to warm-start**

Thank You